IDENTIFIKASI POLA WARNA CITRA GOOGLE MAPS MENGGUNAKAN JARINGAN SYARAF TIRUAN METODE LEVENBERG –MARQUARDT DENGAN MATLAB VERSI 7.8

Tulus Bangkit Pratama[1], John Adler[2]
[1,2] Jurusan Sistem Komputer Unikom, Bandung
[1]bangkitpratama@rocketmail.com, [2]johnadler2zz@yahoo.com

ABSTRAK

Riset-riset mengenai Jaringan Syaraf Tiruan telah banyak diimplementasikan dalam segala bidang ilmu, kuhususnya dalam bidang ilmu pengetahuan alam atau Sains. Jaringan Syaraf Tiruan (JST) merupakan mesin yang didesain untuk memodelkan cara yang digunakan otak untuk mengerjakan suatu fungsi tertentu. Melihat perkembangan mengenai Jaringan Syaraf Tiruan muncul ide untuk mengaplikasikan Jaringan Syaraf Tiruan ke dalam bidang ilmu Geografi mengenai identifikasi pola warna citra peta untuk membedakan kenampakan geografis pada peta, pengaplikasiannya diterapkan pada peta Google Maps, sebuah jasa peta globe virtual gratis dan online yang disediakan oleh Google yang dapat diakses di http://maps.google.com.

Dalam identifikasi pola warna citra dilakukan proses pengolahan citra,yaitu kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik pengolahan citra yang digunakan adalah teknik Segmentasi. Metode training yang digunakan pada penelitian ini menggunakan algoritma Levenberg-Marquardt.

Dari hasil penelitian pada proses training diperoleh nilai epoch sebanyak 17 iterations dari batas maximum 100 iterations. Sedangkan waktu yang digunakan selama proses training sebanyak 6 detik, nilai performance = 2,06.10^7*, nilai gradient = 8,49.* 10^5*, Mu = 1,00.* 10^4*, validation check = 6 dan nilai regresi (R) untuk proses training sebesar 0.95889 dan nilai regresi (R) dari seluruh proses secara berturut-turut adalah 0.91483, nilai regresi tersebut menunjukan adanya hubungan kelinieran antara output jaringan dengan target. Identifikasi pola warna citra pada peta Google Maps menggunakan aplikasi Jaringan Syaraf Tiruan Levenberg Marqurdt telah berhasil membuat pola citra baru yang diidentifikasikan sebagai daerah daratan dan bukan daratan. Secara keseluruhan, aplikasi yang dibuat berhasil memenuhi fungsi yang diinginkan, yaitu untuk membantu penderita buta warna.*

Kata kunci: Jaringan Syaraf Tiruan, Pengolahan Citra, Teknik Segmentasi, Levenberg-Marquardt.

1. PENDAHULUAN

Pada peta dunia terdapat simbol warna untuk kenampakan geografis dan pada setiap warna memiliki makna tertentu. Misalnya kenampakan hipsografi atau relief muka bumi, menggunakan warna dasar coklat, dari coklat muda sampai coklat tua, kenampakan hidrografi atau wilayah perairan (sungai, danau, laut), menggunakan warna dasar biru, dari biru muda (hampir putih) sampai biru tua (kehitaman) dan Kenampakan vegetasi (hutan, perkebunan), menggunakan warna dasar hijau. Warna hijau juga digunakan untuk menggambarkan wilayah dataran rendah.Berdasarkan permasalahan yang diuraikan di atas, maka penulis tertarik untuk mengimplementasikan Jaringan Syaraf Tiruan dengan metode *training Lavenberg-Marquardt* dalam mengidentifikasi pola warna citra *Google Maps* dan membuat aplikasi pengolahan citra untuk mengetahui area daratan dan bukan daratan pada peta *Google Maps*. Dalam penggunaannya diharapkan aplikasi yang dibuat dapat memberikan informasi yang sesuai dengan tujuan atau kepentingan pengolahan citra dan dapat membantu penderita buta warna untuk mengenali objek-objek yang ada di permukaan bumi melalui gambar peta.

2. TEORI PENUNJANG

Google Maps

Google maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google yang dapat diakses di http://maps.google.com. Google Maps memberikan informasi mengenai peta dan pemetaan suatu daerah, selain itu *Google Maps* juga mampu memberikan informasi mengenai kepadatan lalu lintas suatu wilayah, bahkan memberikan pencitraan melalui darat ataupun udara.

Algoritma Levenberg Marquardt

Algoritma *Levenberg-marquardt* merupakan pengembangan algoritma *backpropagation* standar. Pada algoritma *backpropagation*, proses update bobot dan bias menggunakan *Negative Gradient Descent* secara langsung sedangkan, algoritma *Levenberg-Marquardt* menggunakan pendekatan matrik *Hesian* (H) yang dapat dihitung dengan :

$$\boldsymbol{H = J^T J} \quad \text{.....(1)}$$

dimana $\boldsymbol{J^T}$ adalah sebuah *tranpose* dari matrik *jacobian* dan $\boldsymbol{J}$ merupakan sebuah matrik *jacobian* yang berisikan turunan pertama dari *error* jaringan terhadap bobot dan bias jaringan.

$$\boldsymbol{J = \frac{\partial er}{\partial w}} \quad \text{..... (2)}$$

∂er = turunan pertama *error* jaringan.

∂w = turunan pertama bobot jaringan.

Sedangkan gradient (g) dapat dihitung dengan :

$$\boldsymbol{g = J^T e} \quad \text{.....(3)}$$

dimana *e* adalah vektor yang menyatakan semua *error* pada *output* jaringan.
Perubahan pembobot (ΔX) dapat dihitung dengan :

$$\Delta X = [\ \boldsymbol{J^T J + \mu I}\ ]^{-}\ \boldsymbol{J^T e} \quad \text{.....(4)}$$

dimana $\boldsymbol{\mu}$ adalah konstanta *learning* dan $\boldsymbol{I}$ adalah sebuah matrik identitas.
sehingga dapat dihitung fungsi bobot-bobot jaringan dan bias (X) untuk perbaikan pembobot dengan :

$$X = X + \Delta X \quad \text{.....(5a)}$$

$$X = X + [\ \boldsymbol{J^T J + \mu I}\ ]^{-}\ \boldsymbol{J^T e} \quad \text{.....(5b)}$$

Pengolahan Citra

Pengolahan citra merupakan kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Masukannya adalah citra dan keluarannya juga citra tapi dengan kualitas lebih baik daripada citra masukan. Misal citra warnanya kurang tajam, kabur (*blurring*), mengandung *noise* (misal bintik-bintik putih) sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi berkurang.

Segmentasi

Segmentasi merupakan proses partisi gambar digital ke beberapa daerah dengan tujuan untuk menyederhanakan ataupun merubah representasi gambar menjadi sesuatu yang lebih bermakna dan mudah dianalisa. Ada beberapa metoda yang sering digunakan dalam segmentasi citra antara lain : metode *thresholding*, metode *shapebased*, metode *region growing*, dan metode statistik atau juga disebut metode *clustering*.

Citra Biner

Citra biner (*binary image*) adalah citra yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih, 0 dan 1 dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta atau objek atau dalam bentuk angka 0 untuk hitam dan angka 1 untuk warna putih. Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya pada proses analisis citra medis yang menggunakan gambar sinar-x.

Ekstraksi Ciri Orde Dua

Salah satu teknik untuk memperoleh ciri statistik orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Pendekatan ini bekerja dengan membentuk sebuah matriks kookurensi dari data citra, Terdapat berbagai jenis 6 ciri tekstural yang dapat diekstraksi dari matriks kookurensi yaitu *Angular Second Moment, Contrast, Correlation, Variance, Inverse Difference Moment,* dan *Entropy*.

Jaringan Syaraf Tiruan (JST)

JST adalah suatu teknologi komputasi yang berbasis hanya pada model syaraf biologis dan mencoba mensimulasikan tingkah laku dan kerja model syaraf terhadap berbagai macam masukan. Sebagai suatu teknologi komputasi, JST merupakan satu teknik pemrosesan informasi yang menggunakan model kuantitatif. Syaraf biologis mengilhami terciptanya suatu proses komputasi yang identik dengan kerja neuron dalam sistem syaraf manusia. Seperti halnya jaringan syaraf biologis, model matematik JST menghubungkan sejumlah masukan dan keluaran suatu sistem secara adaptif yang diorganisasikan dalam lapisan elemen pemroses seperti layaknya hubungan antar *neuron* syaraf biologis.

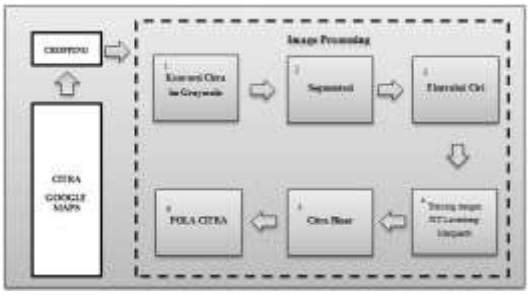
Secara garis besar, proses belajar JST dapat dibagi menjadi 2:

1. JST yang menggunakan paket pelatihan sebagai proses belajar dan dikenal sebagai proses belajar dengan pengawasan.
2. JST tanpa paket pelatihan pada proses belajar dan umumnya disebut proses belajar tanpa pengawasan.

3. PERANCANGAN SISTEM

Diagram Blok Sistem

Berikut merupakan gambar diagram blok sistem yang dibuat (Gambar.1):



Gambar.1 Diagram Blok Sistem

Proses pengambilan citra

Pengambilan *image* input dilakukan dengan cara *cropping* langsung pada *Google Maps* yang dapat diakses melalui situs *https://maps.google.co.id*. Berikut adalah citra asli dari hasil *cropping* pada *Google maps* :



Gambar.2 Citra Asli Google Maps

Pada (Gambar.2)Ukuran *image* input yang akan diolah yaitu *image* yang berukuran 50x512 sampai 384x512 dengan format *image* .jpg.

Sistem Pengolahan Citra

a. Konversi Citra *Grayscale*
   Pada perancangan sistem pengolahan citra dibutuhkan citra *grayscale*, karena citra warna asal merupakan citra RGB sehingga dibutuhkan teknik filter untuk mengubah nilai masing-masing *pixel* menjadi derajat keabuan dengan teknik *Grayscale filter* yaitu mengubah sebuah gambar berwarna menjadi gambar hitam putih dengan cara mengubah efek warna dari masing-masing *pixel* menjadi derajat keabuan.

b. Segmentasi Citra
   Beberapa teknik dan fungsi segmentasi citra yang digunakan pada aplikasi *matlab* adalah teknik *thresholding,* fungsi *entropy* dan fungsi *strel*. Fungsi *entropy* adalah nilai statistik yang dapat digunakan untuk mengkarakterisasi tekstur citra *input*, karena sifatnya yang diskrit (nilai piksel tidak dalam bentuk pecahan) fungsi ini sesuai untuk mencari citra biner yang memiliki nilai ‘0’ dan ‘1’. Selanjutnya teknik *thresholding*, *thresholding* merupakan proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimilikinya. Piksel yang memiliki derajat keabuan lebih kecil dari nilai batas yang ditentukan akan diberikan nilai 0, sementara piksel yang memiliki derajat keabuan yang lebih besar dari batas akan diubah menjadi bernilai 1. Setelah teknik *thresholding* selanjutnya menjalankan fungsi *strel*, proses ini biasanya digunakan untuk kehalusan, pengisian atau menghapus objek dalam suatu citra *grayscale* atau biner.

c. Ekstraksi Ciri Orde Dua
   Salah satu teknik untuk memperoleh ciri statistik orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Pendekatan ini bekerja dengan membentuk sebuah matriks *kookurensi* dari data citra, dilanjutkan dengan menentukan ciri sebagai fungsi dari matriks tersebut.

d. Arsitektur Jaringan Syaraf Tiruan
   Secara garis besar jalannya proses pembuatan arsitektur Jaringan Syaraf Tiruan dari sistem. Arsitektur jaringan syaraf tiruan dari sistem menggunakan arsitektur multilayer network atau jaringan multilapis. Terdapat 3 lapis layer , yaitu :

- Input layer sebanyak 1 neuron
- Hidden layer sebanyak 20 neuron
- Output layer sebanyak 6 neuron

Metode *training* yang digunakan sistem adalah *Levenberg Marquardt learning*. Sistem melakukan pembelajaran terhadap input dengan target yang sudah ditentukan untuk memperoleh nilai-nilai bobot dan bias yang optimum pada tiap unit *neuron hidden layer* dan *output layer*.

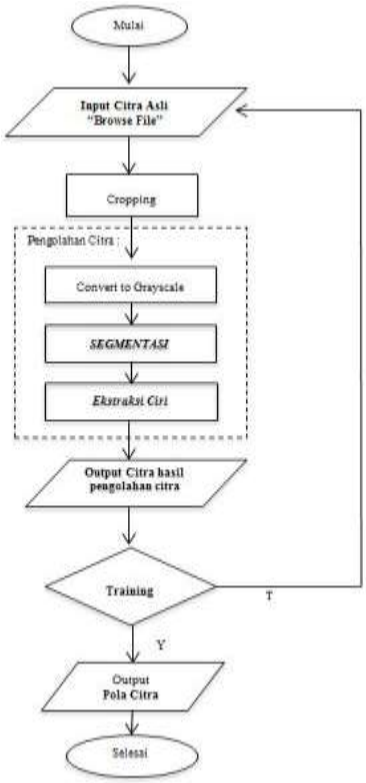
Proses *Training* Levenberg-Marquardt

Dalam rangka mengimplementasikan algoritma *Levenberg-Marquardt* untuk pelatihan jaringan syaraf tiruan, ada satu masalah harus dipecahkan yaitu bagaimana seseorang mengatur pelatihan proses iteratif untuk memperbarui bobot. Pada bagian ini, pelaksanaan proses *training* dengan algoritma *Levenberg-Marquardt* akan disusun sebagai berikut :

a. Menentukan input dan target *training*.
   Nilai input diperoleh dari proses sebelumnya yaitu nilai dari enam ciri statistik orde dua diantaranya Angular Second Moment (CiriASM), Contrast (CiriCON), Correlation (CiriCOR), Variance (CiriVAR), Inverse Difference Moment (CiriIDM), dan Entropy (CiriENT).
b. Pengambilan sampel sebanyak enam kali.
   Dilakukan dengan cara *cropping* dari *image* hasil pengolahan citra. Kriteria sampel yang diambil adalah dua kali *cropping* daerah yang bernilai ‘0’(warna hitam), dua kali *cropping* daerah yang bernilai ‘1’ (warna putih) dan dua kali *cropping* daerah yang bernilai ‘0’ dan ‘1’ (warna hitam dan putih). Proses ini bertujuan untuk menentukan enam nilai ciri statistik orde dua dari citra yang akan di *training* melalui perhitungan ciri orde dua.
c. *Training* citra.
   Dengan menggunakan jaringan syaraf tiruan yang telah dibuat, sampel di *training* menggunakan jaringan syaraf tiruan *Levenberg-Marquardt* (*trainlm).*
d. Klasifikasi pola warna citra.
   Deteksi tekstur pada citra dengan cara mensortir keseluruh bagian *pixel* citra.

Flowchart Perancanngan Aplikasi

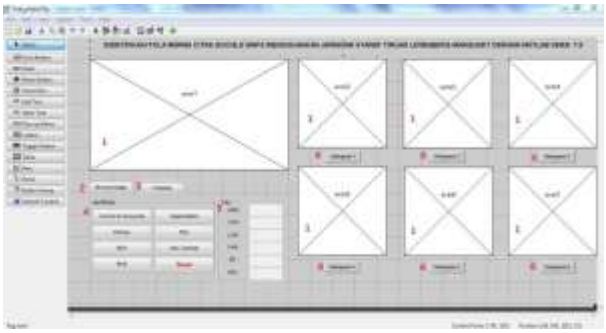
Dengan *flow chart*, proses dari alur program yang dibuat digambarkan dengan sederhana menggunakan simbol yang mudah untuk dimengerti, berikut *flowchart* dari perancangan aplikasi :



Gambar.3 Flowchart Perancangan Aplikasi

Software Antarmuka Sistem

Berikut merupakan tampilan dari *Software* antarmuka sistem Identifikasi Pola Warna Citra Google Maps Menggunakan Jaringan Syaraf Tiruan *Levenberg-Marquadt* Dengan Matlab Versi 7.8.



Gambar.4 Antarmuka Sistem

Berikut penjelasan dari tiap komponen pada sistem yang dibuat : (Gambar.4)

1. *Picture Box,* untuk menampilkan *image*.
2. *Pushbutton* ***Browse Image***, untuk mengambil *File image* yang sudah disimpan.
3. *Pushbutton* ***Cropping***, untuk cropping *image.*
4. **Panel Identifikasi**, didalamnya terdapat menu-menu proses pengoahan citra diantaranya :
   a. Pushbutton ***Convert to Grayscale***
   b. Pushbutton ***Entropy*** **BW1**
   c. Pushbutton **BW2**
   d. Pushbutton ***Segmentation***
   e. Pushbutton **Fitur**
   f. Pushbutton **Atur** ***Contrast***
   g. Pushbutton ***Reset***
5. Panel Fitur, menampilkan nilai ciri orde dua yang merupakan output dari *pushbutton* **Fitur.**
6. Tombol Histogram, untuk menampilkan histogram dari *image* yang diolah.
   a. *Pushbutton* **Histogram1**: Histogram dari citra hasil *cropping.*
   b. *Pushbutton* **Histogram2**: Histogram dari citra *grayscale.*
   c. *Pushbutton* **Histogram3**: Histogram dari citra *entropy.*
   d. *Pushbutton* **Histogram4**: Histogram dari citra BW1.
   e. *Pushbutton* **Histogram5**: Histogram dari citra BW2.
   f. *Pushbutton* **Histogram6**: Histogram dari citra hasil segmentasi.

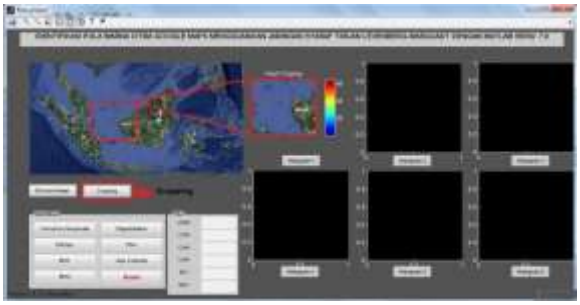
4. Hasil Pengujian

Langkah-langkah pengujian yang dilakukan yaitu setelah program Matlab dijalankan,

Pengujian Program Aplikasi Pengolahan Citra

Berikut adalah langkah-langkah pengujian program aplikasi pengolahan citra yang telah dibuat, yaitu :

1. Open File, tekan pushbotton *Browse Image* untuk menampilkan citra asli. Citra asli adalah citra yang sebelumnya sudah di capture dari *Google Maps* dan dsimpan ke dalam folder file.
2. Lakukan *Cropping* pada citra asli pilihlah daerah mana yang akan dicari karakteristiknya sebagai sampel citra yang akan diolah,lakukan dengan menekan pushbutton *Cropping*. Tekan *pushbutton* Histogram1 jika ingin melihat histogram dari hasil *cropping.*

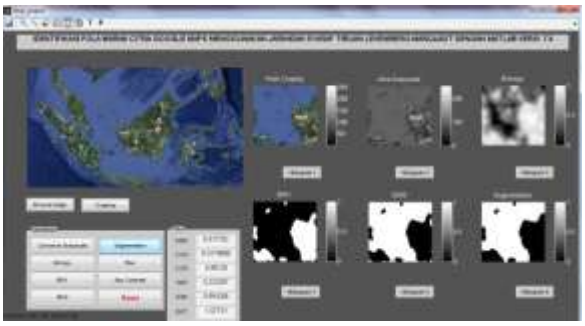


Gambar 5. Cropping citra

3. Tekan *pushbutton Convert to Grayscale* untuk konversi citra RGB menjadi citra *Grayscale*. Tekan *pushbutton* Histogram2 jika ingin melihat histogram dari citra grayscale.
4. Tekan *pushbutton Entropy* untuk mengkarakterisasi tekstur citra grayscale. Tekan pushbutton Histogram3 jika ingin melihat histogram dari citra hasil entropy..
5. Tekan *pushbutton* BW1, pada proses ini digunakan teknik *thresholding* untuk mendapatkan citra biner. Tekan *pushbutton* Histogram4 jika ingin melihat histogram dari citra BW1.
6. Selanjutnya, tekan *pushbutton* BW2 untuk menentukan mana daerah daratan dinyatakan oleh nilai ‘0’ dan mana daerah bukan daratan dinyatakan dengan nilai ‘1’. BW2 merupakan negasi dari BW1 (~BW1). Tekan *pushbutton* Histogram5 jika ingin melihat histogram dari citra BW2.
7. Tekan *pushbutton Segmentation* untuk penghalusan citra biner. Tekan pushbutton Histogram6 jika ingin melihat histogram dari citra hasil segmentasi.
8. Tekan *pushbutton* Fitur untuk menampilkan nilai ciri orde dua.

Dari langkah-langkah pengujian diperoleh keluaran, diperlihatkan pada Gambar.6. , semua *pushbutton* telah berfungsi dengan baik. Pengujian dilakukan dengan mengambil sampel citra dari salah satu bagian citra asli yang dilakukan melalui proses *cropping* kemudian sampel citra di olah melalui proses pengolahan citra sehingga didapatkan citra biner sebagai

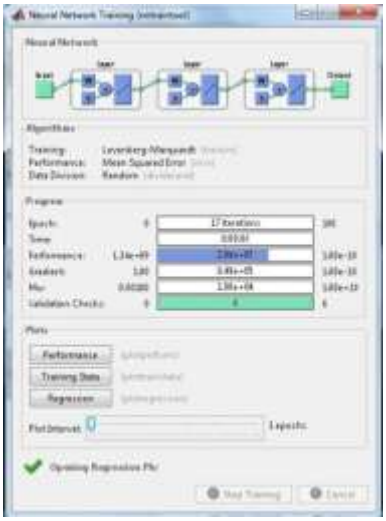
output dari proses pengolahan citra tersebut. Dari citra output yang didapat. Langkah selanjutnya adalah proses *training* citra output hasil dari pengolahan citra.



Gambar 6. Antarmuka Hasil running Aplikasi Pengolahan Citra

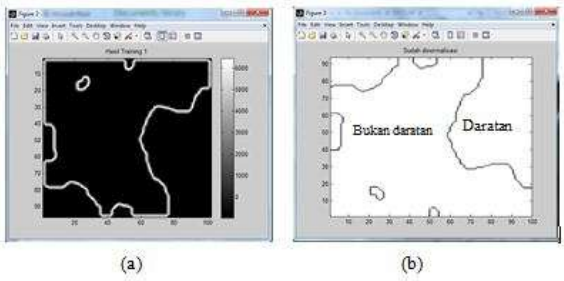
Training citra biner menggunakan jaringan syaraf tiruan *levenberg-marquardt*

Caranya *running list* program `trainLM.m` pada *menu editor* matlab. List program `trainLM.m` sebelumnya sudah dibuat dan di simpan pada *file folder*. Lalu ambilah sampel sebanyak enam kali dengan ketentuan kriteria sampel yang diambil adalah dua kali *cropping* daerah yang bernilai '0'(warna hitam), dua kali *cropping* daerah yang bernilai '1' (warna putih) dan dua kali *cropping* daerah yang bernilai '0' dan '1' (warna hitam dan putih). Setelah itu muncul *window nntraintool.*



Gambar 7. Window nntraintool

Pada gambar 7. terlihat pada panel *progress* hasil dari *training* citra diatas, diperoleh nilai *epoch* sebanyak 17 *iterations* dari batas *maximum* 100 *iterations.*, waktu yang digunakan selama proses *training* sebanyak 6 detik, nilai *performanc*e = $2{,}06.10^{7}$, nilai *gradient* = $8{,}49.\ 10^{5}$, Mu = $1{,}00.\ 10^{4}$ dan *validation check* =6. Langkah pengujian selanjutnya mengklasifikasi pola warna citra, dilakukan untuk mengidentifikasi karakteristik pola warna citra dengan cara mensortir keseluruh bagian *pixel* citra untuk mendeteksi adanya tekstur pada citra.



Gambar 8 Citra hasil training (a). Sebelum dinormalisasi (b). Sudah dinormalisasi

Pada gambar diatas, proses identifikasi membutuhkan waktu 3 menit 10 detik untuk menyapu ke semua daerah citra dengan ukuran citra 92 x 102 *pixel*. Dengan demikian bentuk pola yang diperoleh dapat diklasifikasikan dimana daerah daratan berada di dalam garis dan yang berada diluar garis bukan daerah daratan.

Analisa

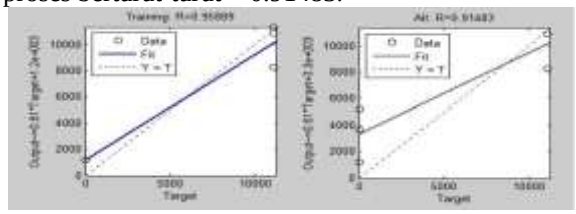
Secara desain dan hasil, jaringan syaraf tiruan yang dirancang memang belum mencapai hasil yang terbaik. Pada Gambar.7 terlihat pada panel *progress* hasil dari *training* citra diperoleh nilai *epoch* sebanyak 17 *iterations* dari batas *maximum* 100 *iterations.*, waktu yang digunakan selama proses *training* sebanyak 6 detik, nilai *performanc*e = $2{,}06.10^{7}$, nilai *gradient* = $8{,}49.\ 10^{5}$, Mu = $1{,}00.\ 10^{4}$ dan *validation check* =6.

Dari batas *maximum* 100 *epoch* didapat 17 kali iterasi maksudnya pada proses *training* terjadi 17 siklus yang melibatkan seluruh pola data *training*, sebenarnya dalam proses *training* memerlukan banyak iterasi sampai mencapai batas *maximum epoch.* sehingga jumlah epoch yang telah ditentukan belum cukup mendapat hasil *performance* jaringan syaraf tiruan yang optimal.

Selanjutnya dalam hasil *training* Jaringan Syaraf Tiruan diperoleh plot regresi, yang menunjukkan hubungan antara *output* jaringan

dan target. Jika pelatihan yang sempurna, *output* jaringan dan target akan persis sama, tetapi hubungan tersebut sangat sulit untuk disesuaikan dan harus berulang – ulang dalam proses *training* sampai nilai *output* mendekati nilai target.

Hasilnya ditunjukkan pada Gambar 9. Nilai R merupakan indikasi hubungan antara output dan target. Jika R = 1, menunjukkan bahwa adanya hubungan linear yang tepat antara output dan target. Jika R mendekati nol, maka tidak ada hubungan linear antara output dan target. Dalam pengujian kali ini, data pelatihan menunjukkan kecocokan. Dalam pengujian kali ini, data pelatihan menunjukkan kecocokan, karena nilai R pada proses *training* mendekati "1" terlihat pada (Gambar. 9) Nilai R *training* = 0.95889 dan nilai R dari seluruh proses berturut-turut = 0.91483.



Gambar 9. Plot Regression

Proses selanjutnya adalah proses mengklasifikasi pola citra dengan cara mensortir keseluruh bagian *pixel* citra untuk mendeteksi adanya tekstur pada citra, di dapatkan pola citra seperti pada (Gambar.8).

Terlihat pada (Gambar.8), citra hasil *training* membentuk pola yang dapat diklasifikasikan dimana daerah yang berada didalam garis merupakan daerah daratan dan yang berada diluar garis merupakan daerah yang bukan daratan. Berdasarkan pengujian hasil ini belum cukup maksimal karena hanya garis tepi saja yang terdeteksi.

Selain itu dalam pengujian proses menyapu ke semua daerah citra dibutuhkan waktu 3 menit 15 detik dengan ukuran citra 92 x 102. Hal tersebut menjadi pertimbangan akan kinerja dari sistem program yang telah dibuat, karena prosesnya masih memakan waktu yang cukup lama, telah dilakukan pengujian sebanyak 5 kali untuk mengetahui pengaruh apa saja yang menyebabkan membutuhkan waktu yang cukup lama . Berikut data hasil percobaan.

Tabel 1. Data Klasifikasi Citra

Percobaan ke-	Ukuran Citra (pixel)	Waktu yang dibutuhkan
1	92 x 102	3 min 15 sec
2	77 x 93	2 min 45 sec
3	58 x 67	1 min 20sec
4	89 x 98	3 min 35 sec
5	126 x 145	11 min 56 sec

Pada tabel 1 hasil percobaan yang telah dilakukan memperlihatkan bahwa ukuran citra *input* mempengaruhi lamanya waktu proses klasifikasi, semakin kecil ukuran citra *input* maka semakin sedikit waktu yang dibutuhkan, sebaliknya jika ukuran citra *input* semakin besar maka semakin banyak waktu yang dibutuhkan.

5. SIMPULAN DAN SARAN

Kesimpulan

Dari perancangan, pengerjaan, dan analisa yang dilakukan selama pembuatan tugas akhir ini, dapat diambil beberapa kesimpulan sebagai berikut:

1. Identifikasi pola warna citra pada peta *Google Maps* menggunakan Aplikasi Jaringan Syaraf Tiruan *levenberg Marqurdt* telah berhasil membuat pola citra baru yang diidentifikasikan sebagai area daratan dan bukan daratan.
2. Dalam proses *training* menggunakan Jaringan Syaraf Tiruan harus dilakukan berulang – ulang sampai nilai *output* mendekati nilai target. Hasil tersebut menunjukan bahwa ada kecocokan antara output dari hasil *training* dengan nilai target.
3. Untuk teknik pengolahan citra *Matlab*, sebagai pilihan akhir digunakan teknik segmentasi citra.
4. Aplikasi pengolahan citra menghasilkan citra biner dari proses segmentasi pola warna citra peta *Google Maps.*
5. Ukuran citra dengan *pixel* tinggi membutuhkan waktu lebih lama untuk diidentifikasi dibandingkan dengan ukuran citra dengan *pixel* yang rendah.
6. Secara keseluruhan, aplikasi yang dibuat berhasil memenuhi fungsi yang diinginkan.

Saran

1. Pada aplikasi yang sudah dibuat, kedepannya diharapkan ada pengembangan aplikasi ini, misalnya untuk identifikasi daerah laut dalam dan rendah.
2. Untuk pengembangan aplikasi pengolahan citra banyak metode pengolahan citra untuk pengenalan pola citra misalnya dengan metode *watershed* .

7. DAFTAR PUSTAKA

[1] H. Demuth, M. Beale, *Neural Network Tool*, Edisi 6, 2000.

[2] Adler, J. Pengaruh Matriks dan Porositas Batuan Karbonat Formasi Parigi. *Majalah Ilmiah Unikom Vol.9,* hal 51-60, Maret 2011.

[3] Prasetyo,E. Pengolahan Citra Digital dan Aplikasinnya Menggunakan Matlab, Andi, Yogyakarta, 2011.

[4] Rahmat. dkk. Pengenalan Pola Sinyal EKG menggunakan JST dengan Algoritma Levenberg – Marquardt. *Seminar Nasional Pasca Sarjana V ITS Surabaya*. 2005.

[5] Gonzales, R.C., and Woods, R.E, *Digital Image Processing Second edition*. New Jersey: Prentice Hall. 2002.

[6] Gunawan, T. *Fakta dan Konsep Geografi untuk SMA/MA kelas XII*, Inter plus, Jakarta, 2007.

[7] Artikel non-personal, 9 Februari 2013, Buta Warna, Wikipedia Bahasa Indonesia, http://id.wikipedia.org/wiki/Buta_warna, diakses 28 februari 2013.